

Bringing Debian APT to the iPhone

The iPhone (or iPod Touch) is a 667MHz computer (albeit one that is only running at 412MHz) with 128MB of RAM and between 4 and 32 GB of flash. For software, it is running a pared down Mac OS X with its standard complement of a FreeBSD-based userland over a Darwin kernel. While some people wonder why anyone would attempt to use it as a Unix workstation, to me and many others it seems ludicrous *not* to.

BSD Subsystem

It is little wonder, then, that one of the first packages that was available for the device was "BSD Subsystem": a set of standard Unix utilities to make your average developer feel right at home. From grep and sed to vim and pico, this package provides the base of many a late-night coding session.

Unfortunately, actually using your iPhone in this fashion for even a short period of time quickly becomes infuriating due to a number of oversights in the configuration and compilation of these fundamental utilities. While it was commendable that someone managed to accomplish this at all, the work shouldn't have stopped at "it installed" and should have continued until "it works". Here is a list of the issues that I've personally had to deal with:

- bash - commands are truncated (and executed) before they have been fully typed
- chmod/chown - always exits with the message "memory exhausted"
- nc/ping - fails to resolve some hostnames and claims "Unknown server error"
- netstat - does not show any useful information on local domain sockets
- passwd - damages the BSD /etc/passwd file, breaking SpringBoard on 1.1.3
- screen - unable to find common terminal types such as "linux" using terminfo
- su - can't run on 1.1.3+ as it is a setuid binary with a relative library reference
- uptime - only says "couldn't get boot time: No such file or directory"
- vim - syntax highlighting is absent and backspace *crashes* the editor

Of these, only chmod and chown have been fixed in the public "BSD Subsystem" package, and only due to the extreme pressure of Apple's 1.1.3 firmware update (which caused all software to be run as mobile instead of root, in turn pushing a requirement to be able to change the ownership of files and mark some applications setuid root).

All of the other problems, though, are also easily correctable if one is willing to spend the time to do so; but, left in their current state, cause a miserable loss of productivity. I *still* find myself hesitating as I hit backspace in vim after having spent a month doing development using this broken copy. Someone just needs to do the work.

CoreOS - iPhone Unix

A few of the remaining issues have already been addressed by an iPhone hacker named [core](#). In the [Unix Tools](#) section of his website he has long provided better copies of such programs as chmod and chown (although his passwd is still broken), and under [Network Tools](#) he has worked hard on fixing some of the issues with tools like wget and ping.

To install core's generally improved copies, however, requires finding, downloading, and installing its modified files separately and manually: a task which might seem alright once, but after the third or forth jailbreak seems downright tedious. People also often do not find out about updates from core, leading to differences in deployed binaries that are difficult to track the behavior of.

Finally, core often doesn't have time to post information about how he performs his magic and provides neither modifications he made to the source code of the packages he has built nor the commands he used to build them on his website, making it impossible to know just what was fixed, or to fix something else without having to redo the work he has already done.

An Open Alternative

In order to provide some relief for these issues, I have decided to start a project called Telesphereo with the goal of creating a distribution of GNU and BSD's userspace for the iPhone as a collaborative, open-source project. The name is an ancient Greek word meaning "to bring fruit to perfection or maturity", which I feel is what needs to be done to Apple's product: it's passable as a phone, but as a portable workstation it is almost unmatched... with the right software.

For distribution, I have chosen Debian's APT, which I have ported to this new platform along with a few custom tweaks to integrate better with Apple's network settings. I have also spent the time to write a UIKit front-end to APT called Cydia, in



Recent Articles

[Yet Another Android Master Key Bug](#)
[Android Bug Superior to Master Key](#)
[Exploit \(& Fix\) Android "Master Key"](#)
[Exploiting a Bug in Google's Glass](#)
[Where did my iOS 6 TSS data go?](#)
[When "Dumb Pipes" Get Too Smart](#)
[Caching Apple's Signature Server](#)
[Tail Call: The New Twitter @Reply](#)
[Debian & Android Together on G1](#)
[iPhone Theming on WinterBoard](#)
[Bypassing iPhone Code Signature](#)
[How to Host a Cydia Repository](#)
[iPhone Applications in Python](#)
[Upgrading the iPhone Toolchain](#)
[Bringing Debian APT to the iPhone](#)

**On iPhone
Development:**

order to make the transition easier for a larger number of users. (Cydia pomonella is the scientific name for the codling moth, which is what we often think of as the stereotypical apple worm. I felt the name was fitting.)

To bootstrap this effort, I have already ported well over a hundred projects: everything in scope from bash to xeyes, along with a number of supporting libraries and scripting languages (including more functional copies of Ruby and Python than are currently available via Installer). All of the source modifications I have had to make, as well as the build scripts for compiling it all, are available via its [subversion repository](#).

While not everything works, what I have done so far is definitely an improvement over what is currently available via BSD Subsystem, and only in a few situations are worse than what core is providing (a problem which will hopefully be corrected quite quickly as core has said that he's up for contributing). When issues *are* found, I encourage people to file defect tickets using [the project's Trac site](#).

If anyone would like to come on board and become the maintainer for one of the existing packages, or help provide ports of more applications, please join the mailing list. While I am currently the only contributor to new distribution, I certainly do not wish this to be the case (both due to lack of expertise in many application areas and a general inability to do everything at once).

Installation Process

From a user perspective, one need only add <http://aptapp.saurik.com/> as a source to Installer and install the package "Cydia Installer" from the "System" category. You might also consider temporarily disabling the auto-lock feature on your device during the installation, as it will take a few minutes to complete.

During the installation, you can expect the following actions to take place:

- **removal of the older files from BSD Subsystem**

In order to replace BSD Subsystem it first has to be removed. Unfortunately, BSD Subsystem is not capable of being "uninstalled", so Cydia Installer simply deletes all the files that BSD Subsystem put on your device out from under it. Please understand that this is not a big deal as, if you somehow absolutely needed BSD Subsystem back, you could always go into Installer and "Reinstall" it.

- **reorganization of your filesystem to free space**

In order to simplify the process of upgrading the iPhone's firmware without the loss of personal data, Apple has organized the disk space on these devices into two partitions, mounted at / and /private/var (which is symbolically linked from /var). The former partition only 300MB large, and is mostly taken up by Apple's software. Unfortunately, it is also the primary location where third party applications are installed.

To help with this problem, if Cydia's installation detects enough disk space on your other partition (where the rest of your 4-32GB of space is allocated) and verifies that these files haven't already been moved by a utility like BossTool it will go ahead and transfer all of your Applications, Wallpapers, Fonts, Ringtones, and Shared data over to the larger partition, freeing almost 150MB of space.

- **installation of Cydia and a base Telesphoreo**

This step, which uses around 35MB of space once complete, will provide a reasonable subset of Telesphoreo that is capable of installing more packages, upgrading what's already installed, performing core administrative tasks like running a terminal, and supporting other applications that previously relied on BSD Subsystem (given a list of requirements provided by myself and my cohort of testers).

Once this process is complete, all further upgrades to both Cydia and all of Telesphoreo will be done using Cydia/APT, so you should occasionally go into Cydia and refresh its repository information to verify that you have the latest software. More on this will be explained in the section on Usage.

If anything goes wrong during this process, please either [file a bug report](#) (you will need to register and login to the site first), come to #iphone on irc.saurik.com (which I am often on, but check at least once a day even if I'm out), or e-mail me for help. (Please only e-mail me directly if neither of the former two options seem appropriate, as the first two options provide the ability for someone else to help you if they are available, which is a more efficient usage of everyone's time.)

"Fake" BSD Subsystem

Note that there is *no* reason to install BSD Subsystem on your device before you install Telesphoreo; and, if you attempt to install it afterwards you will *damage* the installation (it will overwrite your new system with its older software). You also may not *upgrade* BSD Subsystem, as this is the same as installing it as far as downgrading key software components is concerned.

If you have just jailbroken your phone you should therefore install Cydia Installer *in place of* BSD Subsystem. If a package does require something Cydia Installer doesn't

place of BSD Subsystem. If a package does require something Cydia installer doesn't provide by default you can always install that one feature later using Cydia.

This basic conflict with the BSD Subsystem package causes two further issues that users need to be aware of. The first is that upgrades made to BSD Subsystem will be listed in Updates, which provides the possibility of accidental reintroduction of the older software. Secondly, some packages *require* BSD Subsystem to be on your phone before they allow their own installation to continue.

Hopefully, BSD Subsystem will slowly be retired, and those packages will be updated in coming months to do more intelligent checks (such as for the specific files they need). In the mean time, however, Telesphoreo provides a workaround called "Fake" BSD Subsystem that has the same "bundle identifier" as BSD Subsystem but does not contain any files. Installation of this package will allow applications that forcibly rely on BSD Subsystem's installation to work, as well as suppress Installer's occasional requests that you upgrade the original package.

While I try to keep "Fake" BSD Subsystem's version number in sync with that of the original BSD Subsystem package, there is often a lag of a few days during which Installer may then try to help you upgrade *back* to BSD Subsystem. You need to avoid this and instead wait for a new version of the "Fake" package to squelch this new notification.

Using Cydia

In general, Cydia works very similarly to Installer, allowing people to quickly get up and running with the newer software. The main interface component is a bar of buttons along the bottom that provide access to the different main screens of the application.

▪ Featured

Displayed when Cydia launches, this screen is a website that contains news related to usage of Cydia and Telesphoreo and information about currently interesting packages. Currently it only displays a single page, but soon it will be the browser of the application, providing access to package-specific websites and more detailed information.

▪ Install

Here you may browse through the catalog of packages that you haven't yet installed, either by section (APT's name for Installer's "category") or in a single alphabetical list. (Currently, the section names that packages are in are still in flux, so please bear with the occasional odd name.)

▪ Changes

Most of the time all you are interested in is "are there any updates to packages I have already installed" or "have any new packages been added recently that I might want to install". This screen unifies these two usage patterns into a single dashboard: updates appear at the top, with all installable packages listed below in reverse chronological order.

This second usage is similar to Installer's Recent Packages category, with two important changes. First, you may scroll back arbitrarily far through time, instead of only viewing the last few days. The goal of this modification is that only incredibly hard-core users are really going to check Installer *every* few days without fail and without ever taking a weekend or a vacation, which means that people may miss an exciting new feature.

Secondly, the time that a package is considered "added" is the time you first saw it, not the time it got added to the repository. This is important, as if you add a new repository all of the packages in it should be considered "new". This definition also keeps packages from getting percolated back to the top of the list every time they are modified by the repository owner: only truly new packages should be considered "new".

▪ Uninstall (or Manage)

A more mundane usage of Cydia is to find out what you currently have installed, and either reconfigure or uninstall these packages. For these situations the "Uninstall" screen is called for, currently named such as Telesphoreo does not make use of APT's configuration system and Cydia does not support it. If and when more options are available it will be renamed to "Manage" to better describe its modified focus.

▪ Search

Finally, there are often cases where you have heard of a package but aren't exactly sure what it is called. That, or you would like to find a package that provides some particular feature that would be mentioned in its description. This is where "Search" comes in: typing a string into this blank will search the complete package list by identifier, name, and description and display the resulting matches.

The final interface element is along the top of the screen: a refresh button that causes Cydia to synchronize the local APT package database with the current set of repositories.

Cydia is synchronized the local apt package database with the current set of repositories, and an indication of when an update was last performed. It is my hope that this update process can be streamlined in the future, allowing it to be performed in the background as users are otherwise using Cydia. If this were to happen, this bar would likely be entirely removed.

Managing Sources

The one screen that was present in Installer that doesn't have a direct counterpart (due to the placement of Search) is "Sources". While currently not finished being implemented, the plan is to have sources be administrated from the Settings application on your device rather than cluttering Cydia with this information. This interaction is modelled after Apple's MobileMail application, where accounts are modified seldomly via Settings while being accessed often via Mail.

In the mean time, APT maintains its list of source repositories in a file called `/etc/apt/sources.list` which may be modified to contain new sources (were any to show up before the management feature is done being implemented). There is also a directory `/etc/apt/sources.list.d` which contains separate `.list` files, each of which may be easily installed/uninstalled via methods such as an Installer package.

As a stepping stone to this point, the little circled-i in the upper-left corner of all the screens provides a dialog box reminding users of this lack of functionality, as well as the places they may temporarily add sources in the APT configuration files. (And I am of course sorry for any inconvenience this may cause.)

Why Not Installer?

One thing that I understand is that it is quite reasonable to question the decision to use Debian APT and write Cydia as opposed to just managing all of Telesphoreo's packages using [Installer](#), which has become the canonical software distribution mechanism for the iPhone: Installer is the first, and often only, piece of software installed by virtually every automated jailbreak mechanism and is supported by [a large number of repositories](#), each providing in the high tens of packages.

Unfortunately, Installer is not open source, and while there are plans for it to be so, there is no release schedule setup and the promise is getting stale. Cydia's source code is [already available](#) and other projects (such as [MxTube](#)) are benefiting from access. Hopefully, Cydia itself will soon be able to take advantage of this and begin to get third party patches and contributions.

Therefore, given the open nature of Telesphoreo, it makes no sense to rely on Installer in any central role. Even so, this has been the most controversial aspect of this project: many have asked why I felt the need to create a "new" application distribution system (porting APT) rather than using the existing, widely deployed solution. The fact that it will someday be open source is then cited as a reason to not "fork" the efforts of iPhone developers.

I feel the best response is simply to provide some perspective: Cydia/APT does *not* conflict with Installer. I run both on my phone and have a large amount of software installed via Installer. There are currently no known compatibility issues between applications that expect BSD Subsystem to be installed and devices running Telesphoreo, and no reason to suspect anything major on that front. Finally, *no one* is forcing anyone, packagers and users alike, to start using Cydia *instead* of Installer.

I think one of the core problems is that people seem to believe that Installer is doing something they haven't seen before: providing easy management of installed applications from a number of sources using a single interface. While neither Mac OS X nor Windows, the two most common platforms for iPhone owners to have had previous experience with, have such systems, the concept of package management is anything but new: almost every distribution of Linux and BSD is based around it. There are even third-party solutions available for Mac OS X and Windows such as [Fink](#) (which, incidentally, is also based around APT).

I have therefore written a [quick treatment on package management](#) and the complexities thereof. This article also includes the personal story in the usage of Installer from the perspective of a packager that led mostly to frustration. It is my hope that this will elucidate the issues that come up when one gets into the advanced levels of package management and what this means for attempts to support such an ambitious project as Telesphoreo.

Concluding Remarks

These issues notwithstanding, I believe that Telesphoreo and Cydia both provide great value to the iPhone community. Even if someone disagrees with a specific implementation decision, by gathering information on how to port applications and libraries to this platform in one place we should be able to lower the cost of getting even more quality software onto all of our devices. Here's hoping I'm right!